UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/623,101	07/18/2003	Guillermo Rozas	TRAN-P072	2896

7590 08/04/2008

WAGNER, MURABITO & HAO LLP
Third Floor
Two North Market Street
San Jose, CA 95113

EXAMINER	
PETRANEK, JACOB ANDREW	
ART UNIT	PAPER NUMBER
2183	
MAIL DATE	DELIVERY MODE
08/04/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/623,101	**Applicant(s)** ROZAS ET AL.	
	Examiner Jacob Petranek	**Art Unit** 2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *17 June 2008*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-14* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-14* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

1. Claims 1-14 are pending.

2. The office acknowledges the following papers:

Claims and arguments filed on 6/17/2008.

3. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 6/17/2008 has been entered.

New Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-3, 5-7, 9-12, and 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Larson (U.S. 5,115,500).

6. As per claim 1:

Larson disclosed a method of processing an instruction, said method comprising:

fetching said instruction using a corresponding address from a memory unit (Larson: Figure 2, column 5 lines 34-67 continued to column 6 lines 1-40)(An instruction is fetched from the I-Store 2 using an address from the memory unit (Instruction Address Register, IAR 3).);

wherein a plurality of possible meanings are associated with said instruction stored at said corresponding address by the same processor (Larson: Figure 2, column 4 lines 56-67 continued to column 5 lines 1-12 and column 5 lines 34-67 continued to column 6 lines 1-40)(There are a possibility of a plurality of meanings for each instruction depending on the concatenated address bits. When an instruction is fetched, it has one of two meanings, which are an instruction of type 1 if the lower three order bits aren't '111' and an instruction of type 2 if the lower three order bits are '111.' The objective of the invention is to allow for execution of two or more different machine types on a single processor. Thus, it's obvious to one of ordinary skill in the art that the type 1 and type 2 instructions are executed by the same processor.);

concatenating a portion of said corresponding address to said instruction to form an extended instruction, wherein selection of said portion of said corresponding address for said concatenating is independent of region of said memory unit from which said instruction is fetched (Larson: Figure 2, column 3 lines 52-64 and column 5 lines 34-67 continued to column 6 lines 1-40)(The selection of the three lower end bits is independent of which region of memory the instruction is fetched from. The three bits are selected to determine how the instruction will be decoded regardless of if the instruction came from the memory units that store type 1 instructions or the memory units that store type 2 instructions.), And wherein said concatenation increases a number of instructions in an instruction set (Larson: Figure 2, column 4 lines 56-67 continued to column 5 lines 1-12)(The instruction set architecture is comprised of both the type 1 and type 2 instruction sets. Prior to the concatenation, only a single type of instructions was allowed to execute. However, now with the concatenation, the number of instructions can at the maximum double the number of opcodes available for instructions. Type 1 and type 2 are different instructions, which results in increasing the number of instructions available to the overall instruction set architecture of the processor.); and

Executing said extended instruction, wherein said portion of said corresponding address determines a meaning for said extended instruction from said possible meanings (Larson: Figure 2, column 5 lines 34-67 continued to column 6 lines 1-40)(The instruction is decoded and then executed with one of the possible meanings, which is dependent on the extended instruction formed from the concatenation.).

7. As per claim 2:

Larson disclosed the method as recited in claim 1 wherein said portion is an address bit (Larson: Fig. 2, col. 5, line 34 to col. 6, line 40).

8. As per claim 3:

Larson disclosed the method as recited in claim 1 wherein said portion is a plurality of address bits (Larson: Fig. 2, col. 5, line 34 to col. 6, line 40).

9. As per claim 5:

Larson disclosed a method of handling an instruction, said method comprising:

generating said instruction, wherein a plurality of possible meanings are associated with said instruction stored at a corresponding address by a same processor (Larson: Figure 2, column 4 lines 56-67 continued to column 5 lines 1-12 and column 5 lines 34-67 continued to column 6 lines 1-40)(There are a possibility of a plurality of meanings for each instruction depending on the concatenated address bits. When an instruction is fetched, it has one of two meanings, which are an instruction of type 1 if the lower three order bits aren't '111' and an instruction of type 2 if the lower three order bits are '111.' The objective of the invention is to allow for execution of two or more different machine types on a single processor. Thus, it's obvious to one of ordinary skill in the art that the type 1 and type 2 instructions are executed by the same processor.);

storing said instruction at a particular address in a memory unit such that a portion of said particular address enables determination of a meaning for said instruction from said possible meanings (Larson: Figure 2, column 2 lines 21-54 and column 5 lines 34-67 continued to column 6 lines 1-40.); and

before executing said instruction, fetching said instruction using said particular address from am memory unit and concatenating said portion of said particular address to said instruction, wherein selection of said portion of said corresponding address for said concatenating is independent of region of said memory unit from which said instruction is fetched (Larson: Fig. 2, col. 3, lines 52-64 and col. 5, line 34 to col. 6, line 40)(The selection of the three lower end bits is independent of which region of memory the instruction is fetched from. The three bits are selected to determine how the instruction will be decoded regardless of if the instruction came from the memory units that store type 1 instructions or the memory units that store type 2 instructions.),

And wherein said concatenation increases a number of instructions in an instruction set (Larson: Figure 2, column 4 lines 56-67 continued to column 5 lines 1-12)(The instruction set architecture is comprised of both the type 1 and type 2 instruction sets. Prior to the concatenation, only a single type of instructions was allowed to execute. However, now with the concatenation, the number of instructions can at the maximum double the number of opcodes available for instructions. Type 1 and type 2 are different instructions, which results in increasing the number of instructions available to the overall instruction set architecture of the processor.).

10. As per claim 6:

Claim 6 essentially recites the same limitations of claim 2. Therefore, claim 6 is rejected for the same reasons as claim 2.

11. As per claim 7:

Claim 7 essentially recites the same limitations of claim 3. Therefore, claim 7 is rejected for the same reasons as claim 3.

12. As per claim 9:

Larson disclosed the method of as recited in claim 5 wherein said generating said instruction and said storing said instruction are performed by a compiler (Larson: Column 1 lines 11-29)(Larson disclosed that a compiler must be generated for each new machine. A compiler by definition translates high-level language into object code prior to the execution of a program. Thus, a compiler generates instructions that are executable on a processor. A compiler is also defined as any program that transforms one set of symbols into another by following a set of syntactic and semantic rules. As shown in figure 2, a rule for the processor is that type 2 instructions can only be placed in memory locations ending with '111.' Thus, it's obvious to one of ordinary skill in the art that this is a semantic rule that the compiler of Larson must follow and correctly place all type 2 instructions only in memory locations ending with '111' and place all type 1 instructions at other memory locations. Thus, the compiler also stores instructions in memory places.).

13. As per claim 10:

Claim 10 essentially recites the same limitations of claim 1. Therefore, claim 10 is rejected for the same reasons as claim 1.

14. As per claim 11:

Claim 11 essentially recites the same limitations of claim 2. Therefore, claim 11 is rejected for the same reasons as claim 2.

15. As per claim 12:

Claim 12 essentially recites the same limitations of claim 3. Therefore, claim 12 is rejected for the same reasons as claim 3.

16. As per claim 14:

Claim 14 essentially recites the same limitations of claim 9. Therefore, claim 14 is rejected for the same reasons as claim 9.

17. Claims 4, 8 and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Larson (U.S. 5,115,500), in view of ("390 Principles of Operation"), herein referred to as IBM.

18. As per claim 4:

Larson disclosed the method as recited in claim 1.

Larson failed to teach wherein the plurality of possible meanings include an integer type of instruction and a floating point type of instruction.

However, IBM disclosed wherein the plurality of possible meanings include an integer type of instruction and a floating point type of instruction (IBM: Pages 7-1 to 7-6, 9-1 to 9-4, and 9-8 to 9-9)(The combination results in type 1 instructions being the instructions of the IBM 390 ISA. Thus, one of the plurality of possible meanings could be a integer instruction from the 390 ISA or a floating point instruction from the 390 ISA.).

Larson disclosed two separate types of instruction used, but failed to disclose what types of ISA's are used. Since the Larson patent was produced from IBM, one of ordinary skill in the art would have been motivated to look at IBM ISA's for more information on what types of instructions are supported. Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement the IBM 390 ISA onto the processor of Larson.

19. As per claim 8:

Claim 8 essentially recites the same limitations of claim 4. Therefore, claim 8 is rejected for the same reasons as claim 4.

20. As per claim 13:

Claim 13 essentially recites the same limitations of claim 4. Therefore, claim 13 is rejected for the same reasons as claim 4.

Response to Arguments

21. The arguments presented by Applicant in the response, received on 6/17/2008 are not considered persuasive.

22. Applicant argues "Accordingly, the instruction identified by the three high order address bits "111" is decoded differently than if the same instruction was identified by the three high order address bits other than "111." As a result, to have two different decodings for the same instruction requires the same instruction to be stored in two different locations, e.g., identified by the three high order address bits "111" and other than "111 ." In other words, an instruction stored in a location has only one type. Thus, in order to associate the instruction with a different type requires the instruction to be stored in a different location of a memory component. As such, Larsen fails to teach or suggest fetching an instruction using a corresponding address from a memory unit, wherein a plurality of possible meanings are associated with the instruction stored at the corresponding address by a single processor, as claimed."

This argument is not found to be persuasive for the following reason. The applicant seems to be interpreting the claimed limitation as an instruction stored at a fixed location, such as only for example "11001100," has a plurality of possible meanings. However, such an interpretation doesn't have written description support from the specification. As shown in figure 1, taking element 40 as the example, the instruction stored at “11001100” would only have a single interpretation of meaning 1.

The claimed limitation is instead interpreted in view of the specification to have written description support by having a stored instruction at a corresponding address have a plurality of meanings dependent on the decoded corresponding address. In other words, the meaning of the instruction stored at “11001100” is dependent upon the last two digits looking at element 40. Thus, the instruction in this example has four possible meanings, which are associated with four different instruction addresses.

The claimed limitation is still read on by Larsen because the decoded instruction address will allow for a plurality of meanings as shown in figure 2.

23. Applicant argues “For example, operations of the same sort written in different format, e.g., "type 2 add registers", is to be executed (see Larsen, col. 7, lines 38-40). The instruction for "add registers" for type 2 is decoded based on the binary content "111" of the three high order bits, which is "XX1A32" (see Larsen, col. 7, lines 45-53). The execution code to be placed in to execute register is identical since exactly the same function is specified by both type 1 add reqister's instruction and the type 2 add register's instruction (see Larsen, col. 8, lines 10-13). As a result, the state of the execute register after the type 2 add register's instruction has been successfully decoded as being the same as it was for the type 1 instruction (see Larsen, col. 8, lines 13-18)."

This argument is not found to be persuasive for the following reason. The applicant seems to be arguing that in the situation where the type 1 instruction and type

2 instruction perform the exact same operation at the corresponding address, there isn’t a plurality of meanings since the operation is the same. The examiner agrees that this certainly could be the case where the overlap between meaning 1 and meaning 2 for a corresponding address results in executing the exact same operation.

For the purposes of example, the examiner will take this argument to its logical conclusion. The logical conclusion of this argument would be that the type 1 and type 2 ISA are exactly the same. Thus, for every single instruction fetched by the processor, a scenario as the applicant has stated will occur. For all instructions fetched, the exact same operation will take place for the type 1 meaning and the type 2 meaning. Even taken to this extreme, there are still a plurality of meanings since the first meaning will be executing the first ISA add operation and the second meaning is executing the second ISA add operation. It doesn't matter that the add operations may for example add the same two x-bit registers together and write the result to the same x-bit register. There are still two meanings.

One of ordinary skill in the art however would find this example of interpreting the reference of Larsen as unreasonable. One of ordinary skill in the art would find no reason for Larsen to go through all of the effort of adding extra logic to differentiate between type 1 and type 2 instructions only to have all instructions being mirrors of themselves. Thus, when the instructions aren't mirrors of themselves, Larsen clearly reads upon the claimed limitation by having the type 1 operation and type 2 operation being different at a corresponding address.

24. Applicant argues “Larsen failed to teach “Wherein said concatenation increases a number of instruction in an instruction set.””

This argument is not found to be persuasive for the following reason. The invention combined two different instruction sets into a single instruction set onto the same processor. It doesn’t matter if the type 1 and type 2 share the same opcode for a particular instruction because they are still two individual instructions within the overall instruction set architecture. This can be shown in figure 3 that shows the instructions of type 1 and type 2 instructions are of different bit sizes. Therefore, the instructions for each the type 1 and type 2 contribute to the total number of instructions for the overall instruction set architecture of the processor. The instructions are encoded differently because Larsen states the processor is to execute from two or more different machine types, this being instructions that are encoded differently.

25. Applicant argues “Larsen failed to teach wherein said generating said instruction and said storing said instruction are performed by a compiler” for claim 9.

This argument is not found to be persuasive for the following reason. Larson disclosed that a compiler must be generated for each new machine. A compiler by definition translates high-level language into object code prior to the execution of a program. Thus, a compiler generates instructions that are executable on a processor. A compiler is also defined as any program that transforms one set of symbols into another by following a set of syntactic and semantic rules. As shown in figure 2, a rule for the processor is that type 2 instructions can only be placed in memory locations ending with ‘111.’ Thus, it’s obvious to one of ordinary skill in the art that this is a semantic rule that the compiler of Larson must follow and correctly place all type 2 instructions only in memory locations ending with ‘111’ and place all type 1 instructions at other memory locations. Thus, the compiler also stores instructions in memory places.

Conclusion

The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jacob Petranek whose telephone number is 571-272-5988. The examiner can normally be reached on M-F 8:00-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner’s supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Eddie P Chan/ Supervisory Patent Examiner, Art Unit 2183	Jacob Petranek Examiner, Art Unit 2183